RailSLAM - Localization of Rail Vehicles and Mapping of Geometric Railway Tracks

Oliver Heirich, Patrick Robertson, Thomas Strang

***Abstract*— The avoidance of train collisions is vital for human safety in railway transportation. Technical approaches are general train control or collision avoidance systems as well as semi-automated or fully autonomous trains. These systems rely on robust and exact train localization as well as an accurate map of the track network. We present Simultaneous Localization and Mapping relying exclusively on train-side sensors. RailSLAM, implemented as a probabilistic filter, uses measurements from multiple sensors and computes a track map. We rely heavily on sensors that are not affected by the harsh environmental conditions often experienced in this application, in particular a low-cost MEMS Inertial Measurement Unit (IMU). Rail vehicle localization methods based on these sensors require a dedicated map with detailed geometric track features in combination with the topological track connections. If this feature map does not exist apriori, it needs to be created. If it does, it may suffer from incompleteness, insufficient accuracy or outdated information. RailSLAM addresses the creation and maintenance of this special track map by a simultaneous estimation of the probabilistic geometric-topological feature-rich track map and the train state. A first proof of concept implementation of mapping is given based on the use of an Extended Kalman Filter with measurements from Global Navigation Satellite System (GNSS) and an IMU.**

I. INTRODUCTION

Safety in railway transportation relies critically on collision avoidance strategies and is addressed by general train control or collision avoidance systems. Railway collision avoidance systems are either designed as a safety overlay system for train driver assistance [1] or integrated in automated train control. The design of collision systems results in a trade off between safety critical missed collision detections and unacceptable false alarms. Key to achieving both low missed detection and false alarm rates is a highly reliable and exact vehicle localization in the track network. Furthermore exact localization is essential for semi-automated train maneuvering or fully automated, autonomous train driving. Generic approaches based solely on Global Navigation Satellite Systems (GNSS) are not reliable in tunnels, urban environments or forests and are not sufficiently accurate for a parallel track scenario as relevant distances are within GNSS accuracy range [2]. Another approach is to deploy costly track-side infrastructure-based positioning systems, such as the Eurobalise [3].

A special, in fact defining property of trains is the strong physical constraint of train kinematics by the rails. It is possible to measure the train kinematics caused by the constraints and feature-rich geometry of track. A localization method based on these sensors requires a dedicated map with geometric track features in combination with the topological track connections (Figure 1). This feature map usually does not exist apriori and need to be created. Once available, it may suffer from incompleteness, insufficient accuracy or outdated information.

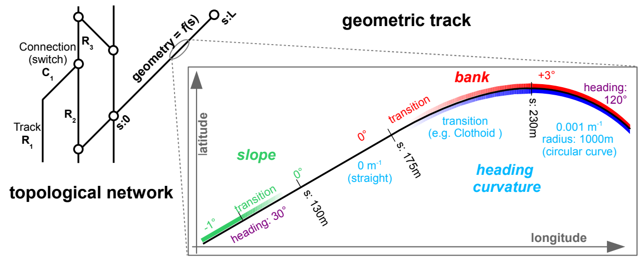

Fig. 1. Topological track network and geometric track map.

We present a simultaneous localization and mapping approach for railways, which we have called RailSLAM, that jointly estimates the train state and the track map based on *intrinsic* and *extrinsic* multi-sensor measurements. It creates the map initially, it validates whether the map is still correct, it extends the map and keeps the map up to date. In normal use, the map is used to perform localization. The collected map data can also be used for physical track maintenance purposes.

Railway positioning needs to work under all environmental conditions. Visibility in the visual and infrared regions can be reduced to zero under some conditions. Other factors affecting sensor reliability and cost effectiveness are susceptibility to mechanical damage (especially for sensors mounted undercarriage) and even theft. We therefore distinguish between two main groups of onboard train sensors: *intrinsic* inner train state sensors and *extrinsic* environment perception sensors. In analogy to human perception, there is proprioceptive sensing (e.g. sensing body position and motion) and exteroceptive sensing (e.g. vision). An example for intrinsic SLAM without exteroceptive sensors for indoor navigation is [4]. The differentiation between intrinsic and extrinsic plays a role in the general RailSLAM derivation of the probabilistic filter.

Classic robotic SLAM approaches [5] estimates landmarks based on extrinsic measurements from vision or range and bearing sensors while an intrinsic odometry is used as the control input. A first SLAM approach for railways has been developed and published by [6], where a multi-hypotheses EKF-SLAM estimates the track map by efficient probabilistic splines based on GNSS position measurements. Our ap-

O. Heirich, P. Robertson and T. Strang are with DLR (German Aerospace Center), Institute of Communications and Navigation, DLR e.V., Oberpfaffenhofen, Germany {oliver.heirich, patrick.robertson, thomas.strang}@dlr.de

proach focuses on the feature-rich track geometry observed by the intrinsic sensors (mainly the IMU). We propose that GNSS and IMU are practical, robust with relative low sensor and installation costs. Nevertheless, our general approach considers the extensibility by extrinsic sensors, despite their possible limitations.

The objective of the paper is to present a physically motivated Bayesian derivation, starting with a discussion of a suitable dynamic Bayesian network. A first simplified implementation with the intrinsic sensors of GNSS and IMU is presented for the generation of the *probabilistic geometric feature-rich map* parametrized on the topological track position.

II. PROBABILISTIC TRAIN NAVIGATION

For a Bayesian dependency analysis, we define the train states as $\mathbf{T}_k$, at the time step k. The bold notation of the variables represents a random variable. We denote $\mathbf{U}_k$ as train control input and the railway environment is $\mathbf{M}$. The extrinsic environment sensors are $\mathbf{Z}_k^{\mathrm{EX}}$ with their errors $\mathbf{E}_k^{\mathrm{EX}}$ and the intrinsic inner train state sensor measurements are $\mathbf{Z}_k^{\mathrm{IN}}$ with the errors $\mathbf{E}_k^{\mathrm{IN}}$.

A. Dynamic Bayesian Network

A dynamic Bayesian network (DBN) visualizes causal dependencies of effects in a directed acyclic graph [7] and encodes relevant aspects for the probabilistic train navigation. Dependency is shown as arrows from the cause to effect, while randomness between two nodes lacks of a connection. In Fig. 2 we draw the DBN for two time steps of a train running in a railway environment, which is equipped with onboard sensors. From the DBN we can see, that the train

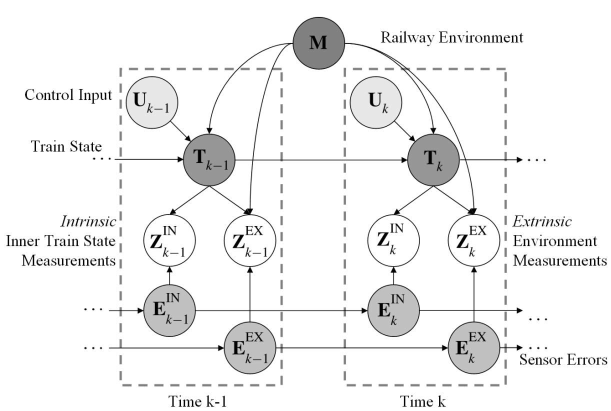

Fig. 2. General dynamic Bayesian network for railway navigation

state $\mathbf{T}_k$ is directly dependent to the last train state $\mathbf{T}_{k-1}$, control input $\mathbf{U}_k$ and the railway environment $\mathbf{M}$. The railway environment represents time-persistent landmarks of any kind and especially the railway tracks, which have a strong influence on the train states caused by physical constraints of the rails. The dependency of $\mathbf{T}_k$ to $\mathbf{T}_{k-1}$ can be explained by the inertia of train. A train can not change its position, speed or attitude randomly between two time steps because of the train mass. The train control input $\mathbf{U}_k$ is random, and so conditionally independent of previous $\mathbf{U}_{k-1}$. $\mathbf{U}$ is the 1D-control by the train driver, which is accelerating or braking and the switch way, which is set automatically or manually by train control or sometimes manually by the train driver on industrial lines.

B. Sensor measurements

The intrinsic $\mathbf{Z}_k^{\mathrm{IN}}$ and extrinsic measurements $\mathbf{Z}_k^{\mathrm{EX}}$ originate from onboard train sensors. $\mathbf{Z}_k^{\mathrm{IN}}$ or $\mathbf{Z}_k^{\mathrm{EX}}$ are conditionally independent on previous measurements $\mathbf{Z}_{k-1}^{\mathrm{IN}}$ or $\mathbf{Z}_{k-1}^{\mathrm{EX}}$.

Intrinsic inner train state sensors $\mathbf{Z}_k^{\mathrm{IN}}$ measure exclusively characteristics of the train such as speed, position, attitude, accelerations and turn rates. Suitable intrinsic sensors are odometry, absolute position and speed vector measurements from GNSS, three dimensional acceleration and turn rate measurements by an IMU. $\mathbf{Z}_k^{\mathrm{IN}}$ is dependent on the train states $\mathbf{T}_k$. With the fact that $\mathbf{T}_k$ is dependent on the railway environment $\mathbf{M}_k$, the railway environment has influence on these measurements through the train states. This can be explained by the constraint of the rails and the track geometry causing kinematic states of the train, which can be measured by the intrinsic sensors. This is considered as the reason why SLAM based only on intrinsic sensors is possible in railways.

Extrinsic environment perception sensors $\mathbf{Z}_k^{\mathrm{EX}}$ measure a characteristic of the surrounding environment of trains. These sensors still depend on some inner train states, as the measurement ranges might depend on train position, attitude and speed. Measurements $\mathbf{Z}_k^{\mathrm{EX}}$ are therefore dependent on $\mathbf{M}$ and $\mathbf{T}_k$. Example sensors are range and bearing sensors by radars or laser scanners, vision sensors by mono or stereo cameras or magnetic sensors. Magnetic sensors measure either the magnetic field as it results from the earth field and nearby magnetic objects or detect changes in an active generated field. Known examples are metal detectors or eddy current sensors for railways [8].

The sensor errors $\mathbf{E}_k^{\mathrm{EX}}$ and $\mathbf{E}_k^{\mathrm{IN}}$ are dependent on their previous time step and therefore characterized by memory behavior. Sensor white noise is not part of $\mathbf{E}_k$, as white noise is random and independent over time and part of $\mathbf{Z}_k$. These errors can be included with special models and are estimated within the SLAM method. The sensor measurements are corrected by these error estimates which allows more precise measurement models. This is considered as an important part of a robust approach for safety related systems with permanent operation all the year round without manual resetting or recalibration and low frequent maintenance.

Examples for intrinsic sensor errors $\mathbf{E}_k^{\mathrm{IN}}$ with memory behavior are calibration errors or multipath and atmospheric effects for GNSS, bias drift for IMUs and wheel slip for odometry.

Examples of extrinsic sensor errors $\mathbf{E}_k^{\mathrm{EX}}$ with memory behavior are calibration errors or errors of a changing environment by objects such other trains or precipitation for range and vision sensors, low illumination or dirty optics for vision sensors and magnetic interference by other trains.

C. Joint posterior and SLAM

The joint posterior is estimating train states $\mathbf{T}_{0:k}$ and sensor errors $\mathbf{E}^{\text{IN}}_{0:k}, \mathbf{E}^{\text{EX}}_{0:k}$ of all time steps 0 to k and the railway environment $\mathbf{M}$, given all the measurements $\mathbf{Z}^{\text{IN}}_{1:k}, \mathbf{Z}^{\text{EX}}_{1:k}$ and all the control inputs $\mathbf{U}_{0:k}$:

$$p(\mathbf{T}_{0:k}, \mathbf{E}^{\text{IN}}_{0:k}, \mathbf{E}^{\text{EX}}_{0:k}, \mathbf{M} | \mathbf{U}_{0:k}, \mathbf{Z}^{\text{IN}}_{1:k}, \mathbf{Z}^{\text{EX}}_{1:k}). \tag{1}$$

The joint posterior contains the map and state estimates and its computation is the goal of SLAM methods. RailSLAM and especially the map is either estimated of intrinsic train kinematic measurements $\mathbf{Z}^{\text{IN}}$ from the influence of $\mathbf{M}$ through $\mathbf{T}_{0:k}$ or of extrinsic measurements by the dependencies of $\mathbf{M}$, $\mathbf{T}$ and $\mathbf{Z}^{\text{EX}}$ or both.

III. BAYESIAN FILTER

The Bayesian filter is derived from the dynamic Bayesian network (Fig. 2) and the joint posterior. The purpose of the derivation is a factorized solution in a recursive form for the Bayesian filter implementation. For a shorter notation, random variables of the same time indices are written: $\{\mathbf{TU}\}_k$ instead of $\mathbf{T}_k$, $\mathbf{U}_k$. Same applies for $\{\mathbf{TE}\}_k$ or $\{\mathbf{TEU}\}_k$. $\mathbf{Z}$ contains all measurements: $\mathbf{Z} : \mathbf{Z}^{\text{IN}}, \mathbf{Z}^{\text{EX}}$. Same applies for $\mathbf{E} : \mathbf{E}^{\text{IN}}, \mathbf{E}^{\text{EX}}$. The factorization of the joint posterior by the product rule results in a map estimation part and a localization part:

$$\begin{aligned} &p(\mathbf{T}_{0:k}, \mathbf{E}^{\text{IN}}_{0:k}, \mathbf{E}^{\text{EX}}_{0:k}, \mathbf{M} | \mathbf{U}_{0:k}, \mathbf{Z}^{\text{IN}}_{1:k}, \mathbf{Z}^{\text{EX}}_{1:k}) = \\ &\quad = p(\mathbf{M} | \{\mathbf{TEU}\}_{0:k}, \mathbf{Z}_{1:k}) \cdot p(\{\mathbf{TE}\}_{0:k} | \mathbf{U}_{0:k} \mathbf{Z}_{1:k}) \\ &\quad = \underbrace{p(\mathbf{M} | \mathbf{T}_{0:k}, \mathbf{Z}^{\text{EX}}_{1:k})}_{\text{mapping}} \cdot \underbrace{p(\{\mathbf{TE}\}_{0:k} | \mathbf{U}_{0:k} \mathbf{Z}_{1:k})}_{\text{localization}} . \end{aligned} \tag{2}$$

The map depends only on the train state and the environment measurements $\mathbf{Z}^{\text{EX}}$ and is conditionally independent of the train state only measurements $\mathbf{Z}^{\text{IN}}$, the control input $\mathbf{U}$ and the sensor errors $\mathbf{E}$. The map is conditionally independent of the environment sensor error $\mathbf{E}^{\text{EX}}$, because with the knowledge of $\mathbf{T}$ as the cause of $\mathbf{Z}^{\text{EX}}$, $\mathbf{E}^{\text{EX}}$ is explained away.

A. General localization posterior

The factorized localization posterior from (2) is derived in the appendix and contains the parts: measurements, error and train transition and the recursive part:

$$\begin{aligned} &p(\{\mathbf{TE}\}_{0:k} | \mathbf{U}_{0:k}, \mathbf{Z}_{1:k}) \propto \\ &\propto \underbrace{p(\mathbf{Z}^{\text{IN}}_k | \mathbf{T}_k, \mathbf{E}^{\text{IN}}_k) \cdot p(\mathbf{Z}^{\text{EX}}_k | \mathbf{T}_k, \mathbf{E}^{\text{EX}}_k)}_{\text{sensor measurements}} \cdot \underbrace{p(\mathbf{T}_k | \mathbf{T}_{k-1}, \mathbf{U}_k)}_{\text{train transition}} \cdot \\ &\quad \cdot \underbrace{p(\mathbf{E}_k | \mathbf{E}_{k-1})}_{\text{error transition}} \cdot \underbrace{p(\{\mathbf{TE}\}_{0:k-1} | \mathbf{U}_{0:k-1} \mathbf{Z}_{1:k-1})}_{\text{recursive part}} \end{aligned} \tag{3}$$

SLAM algorithms compute simultaneously, which means in the same time step, at first the localization with the map from the previous time step and updates then the map with the computed localization. As seen from the DBN in Fig. 2, the map is a cause of the train state $\mathbf{T}$ and the environment measurement $\mathbf{Z}^{\text{EX}}$. A common trick is the integration of the map information by a marginalization [5]. Therefore the estimates are conditionally extended by the map and multiplied by the map estimate of the *previous step*. The integral over the map for the rail vehicle transition is:

$$\begin{aligned} &p(\mathbf{T}_k | \mathbf{T}_{k-1}, \mathbf{U}_k) = \\ &= \int_M \underbrace{p(\mathbf{T}_k | \mathbf{T}_{k-1}, \mathbf{U}_k, \mathbf{M})}_{\text{transition on the map}} \cdot \underbrace{p(\mathbf{M} | \mathbf{T}_{0:k-1}, \mathbf{Z}^{\text{EX}}_{1:k-1})}_{\text{map at k-1}} \, d\mathbf{M}, \end{aligned} \tag{4}$$

and for extrinsic measurements:

$$\begin{aligned} &p(\mathbf{Z}^{\text{EX}}_k | \mathbf{T}_k, \mathbf{E}^{\text{EX}}_k) = \\ &= \int_M \underbrace{p(\mathbf{Z}^{\text{EX}}_k | \mathbf{T}_k, \mathbf{E}^{\text{EX}}_k, \mathbf{M})}_{\text{map dependent measurements}} \cdot \underbrace{p(\mathbf{M} | \mathbf{T}_{0:k-1}, \mathbf{Z}^{\text{EX}}_{1:k-1})}_{\text{map at k-1}} \, d\mathbf{M}. \end{aligned} \tag{5}$$

B. Simplified proof of concept implementation

From the generalized Bayesian train navigation filter, we implement a reduced version with two intrinsic sensors of the type $\mathbf{Z}^{\text{IN}}$ by a GNSS receiver $\mathbf{Z}^{\text{GNSS}}$ and an IMU $\mathbf{Z}^{\text{IMU}}$. The memory based sensor errors $\mathbf{E}$ are neglected for this approach. Figure 3 shows the simplified DBN for the implementation.

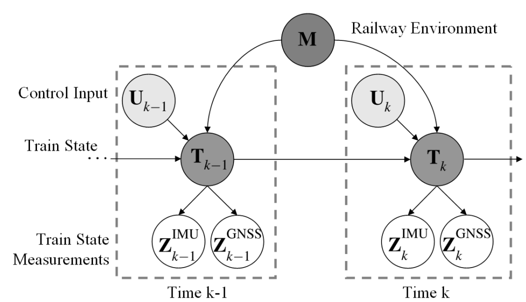

Fig. 3. Simplified dynamic Bayesian network for proof of concept

IV. SLAM FOR RAILWAYS

Two main phases are defined for RailSLAM [9], which are the white space mapping and the SLAM on a prior map phase as well as transitions between these phases, as seen in Figure 4.

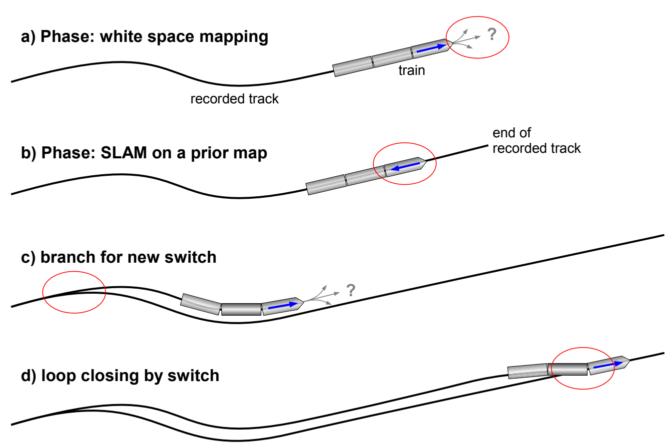

Fig. 4. RailSLAM-phases and transitions

A. White space mapping phase

White space is the situation if there is no information known about the tracks, in analogy to paper maps. In this phase, the train estimates a new track and records the positions and geometry filtered from the sensors. A bayesian filter estimates equation (3) directly. As there is no apriori map, there is no need for an extension to (4), the topological train location is simply the end of the track and the distributions of location dependent states of $\mathbf{T}$ are recorded. Switches are not recorded in the white space phase except for those merging to a known track.

B. SLAM on prior map phase

In Figure 4(b), the train reverses and runs now on a previously recorded track, i.e. where a prior map exists. In the SLAM phase, the train is localized on a map of the previous state and updates the map in the same time step. In this phase the map is enhanced by every revisit. For the localization part, we presented a particle filter approach based on geometric track features [10]. The SLAM phase is valid as long as the train stays on known tracks which have been visited at least once.

C. Phase transitions

In Figure 5 we show the state machine for RailSLAM with the two main states and their transitions.

1) Branch by switch: Figure 4(c) shows the transition from SLAM to white space phase after branching from a known track by a switch.

2) End of map: This transition happens when a train runs from a known map into the white space without a switch.

3) Reverse track: As showed in Figure 4(b), the reverse transition happens from a change of the train travel direction in the white space phase.

4) Circular loop closure: On circular networks, where a train arrives its initial position without changing travel direction, a circular loop closure happens. The phase changes from white space mapping to SLAM by the known initial position. Common SLAM approaches in robotics [5] rely on a loop closure path for a map convergence. Here, the convergence of RailSLAM is not dependent on a loop closure because of absolute GNSS positions. GNSS positioning may fail in some scenarios but there are still enough absolute positions to avoid the need of a loop closure for convergence.

5) Merge by switch: A common situation of railway lines with two parallel tracks with dedicated directions is a loop closure when the tracks merge by a switch. Figure 4d) shows the situation when two parallel tracks merge by a switch.

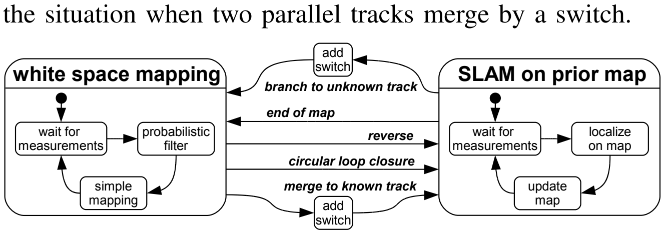

Fig. 5. RailSLAM-phase state machine

D. Unknown data association

As a consequence of limited GNSS position accuracy and availability, there is an ambiguity in the correct track estimation, especially at switches and parallel tracks. RailSLAM has to cope the discrete track ambiguity by phase transition estimates. Additionally an along track uncertainty of the one dimensional track position is present, which results in a two dimensional unknown data association for a correct localization and map update.

V. RAILWAY TRACKS

A. Track Topology

Railway tracks are connected by switches, crossings or diamond switch crossings. We define a track as the vertex between connection nodes, i.e. a track contains no switch or crossing. This definition ensures, that a track is always true one-dimensional with no other access than the track begin or track end. A switch connects three tracks, a crossing four tracks. According to the travel direction and switch way position, a track branches into two other tracks when passing a switch facing and two tracks merge into one track by passing trailing.

B. Topological coordinates

The goal of train localization is to estimate the train position in the track network by topological coordinates. This topological position is defined by a unique track ID R and a track length variable s. The origin of that length has to be defined for direction dir of the train related to the track. A positive direction points away from the origin, a negative towards the origin. The topological pose is a triplet of track ID, length and direction and defines the train position and attitude in topological coordinates unambiguously:

$$\mathrm{P}^{\text{topo}} = \{R, s, dir\}. \tag{6}$$

C. Track Geometry

Railway tracks are fixed to the earth, so any position on the tracks represent as well an absolute geographic position. The geometry of a track at a certain position is given by the attitude and the changes of the attitude over position. The track attitude contains heading ψ, bank ϕ and slope θ. Bank ϕ is the lateral inclination and used in curves to compensate the lateral centripetal acceleration effects in curves. Slope θ is the inclination along a track and can be ascent or decent for changing altitude level. The bank of a track changes continuously by a ramp or a similar transition and is called bank curvature $\frac{d\phi}{ds}$. The slope change over the track position is called slope curvature $\frac{d\theta}{ds}$ and is also continuously. The change of the heading over the track position is the heading curvature $\frac{d\psi}{ds}$.

D. Geometric coordinate frames

We defined the coordinate geometric frames in [10] and revisit the most important definitions:

1) ***Earth Frame****:* Absolute geographic positions is defined in the earth frame by latitude, longitude, altitude (LLA) coordinate system with WGS84 datum.

*2) **Navigation Frame:*** The navigation frame is defined by the Cartesian north-east-down (NED) system. This frame spans an orthogonal plane to the gravity vector at a geographic position and allows position computations in the vicinity of the train or track position. Conversion from LLA to NED frame $\mathcal{C}_{\text{LLA}}^{\text{NED}}$ and back are found in [2].

*3) **Track Frame:*** The geometric track frame is a Cartesian coordinate system with the axes along-track, cross-track and down. The track attitude is defined by right-handed Euler angles slope (θ), bank (ϕ) and heading (ψ) between the local tangent frame and the track frame as shown in Figure 6. The transformation from the track frame to the navigation frame (NED) is defined by the rotation matrices of the angles ψ, θ and ϕ in [11]:

$$\mathcal{C}_{\text{track}}^{\text{nav}} = \begin{pmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{pmatrix}. \tag{7}$$

The s and c in the rotation matrix refer to sine and cosine. The transformation from NED to track frame is $\mathcal{C}_{\text{nav}}^{\text{track}} = \mathcal{C}_{\text{track}}^{\text{nav}\top}$. The rotation axes of the turn velocities $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are defined

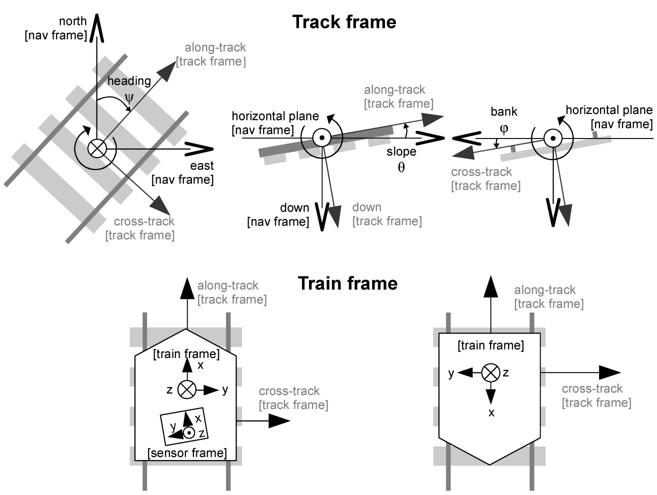

Fig. 6. [Top] Track frame and navigation frame (NED). [Bottom] Train frame, track frame and sensor frame.

in the following way: The $\dot{\phi}$ axis is the along-track axis in the horizontal plane, $\dot{\theta}$ the one to the cross-track axis and $\dot{\psi}$ is aligned with the down axis (see Fig. 6).

*4) **Train Frame:*** The train frame is a Cartesian system with the axes x for along track, y for cross track and z for down. The train frame is directly dependent on the track frame. According a positive ($dir = +1$) or negative ($dir = -1$) train direction the attitudes and the sign of the axes x and y changes. The train motion, or speed $\dot{s}$ is defined in the x-axis of the train frame and can be forward ($\dot{s} > 0$) or backward ($\dot{s} < 0$), as shown in Figure 4b). The train motion direction is the moving train in relation to the track direction definition.

*5) **Sensor Frame:*** A sensor frame defines the attitude and position of a sensor relative to the train frame.

E. Geometric Track Model

The key idea of a topological and geometric feature map is the parametrization of the geometry by the topological, one-dimensional track position s. This geometry is stored in so called track points. A track point Tp_s is a geometric vector of one position s of a track R and is defined as:

$$\text{Tp}_s = \{s, \underbrace{lat_s, lon_s, alt_s}_{\text{geo position}}, \underbrace{\phi_s, \theta_s, \psi_s}_{\text{attitude}}, \underbrace{(\tfrac{d\phi}{ds})_s, (\tfrac{d\theta}{ds})_s, (\tfrac{d\psi}{ds})_s}_{\text{curvatures}}\}. \tag{8}$$

Track data between these points are interpolated by polygonal line approximation, which interconnects points with lines. We favor the simple linear polynomial approximation, because it is easy to implement. More advanced methods for the geographic track representation use spline approximations [12]. In contrast to spline approximations, linear polynomials do not oscillate and do not require special calculations of the spline break points. The drawbacks of linear approximation are linearization errors and to keep them low, a relative high number of track points need to be stored. Our idea is, to treat the different geometry values as spatial geometry signals. These signals are a function of the spatial domain by the one-dimensional track position s and we sample these signals with a sufficient number of track points. The track points are constantly spaced by ΔS over the length L of one track.

Once all the track map data is collected and enforced by multiple runs, a track point thinning strategy with low information loss or data compression can be considered. We define single precision (32bit) values for the track point data, while latitude and longitude is double precision (64bit). Table I shows the data volume estimates for the geometric data by different track point spacings in megabytes for the German and the track networks of 27 European countries. Topological connections, speed limits, or other data are not respected in the table.

TABLE I
UNCOMPRESSED DATA VOLUMES FOR DIFFERENT SPACINGS

Network	Size	Spacing: 1m	5m	10m
Germany	63,839km [13]	2,922MB	584MB	292MB
EU27	330,892km [13]	15,145MB	3,029MB	1,515MB

F. Map database

The geometric track information of a particular position is provided by the digital map and referenced by the topological address of track ID R and position s:

$$\text{Tp}_s = f_{\text{map}}(R, s). \tag{9}$$

In practice, the map of the track network M is organized by a list of tracks:

$$\text{M} = \{\text{R}_1, \text{R}_2, \cdots, \text{R}_i\}, \tag{10}$$

Every track contains a unique track ID R, connections and track data by j track points parametrized to one-dimensional position s with constant spacings of ΔS.

$$\text{R} = \{\text{Tp}_0, \text{Tp}_{\Delta S}, \text{Tp}_{2\Delta S} \cdots, \text{Tp}_{(j-2)\Delta S}, \text{Tp}_L\}. \tag{11}$$

G. Probabilistic map

SLAM maps depend on an uncertainty representation. As Kalman filter based SLAM methods use Gaussian distributions, where as particle filters rely on multiple realizations of the map. A combined Rao-Blackwellized approach uses multiple maps with Gaussians. For a Gaussian uncertainty, variances for every variable of the geometric track data extend the track points and the amount of data is doubled. The result is a *probabilistic geometric feature-rich map*.

VI. RAIL VEHICLE MODEL

A. Train states

As the train has a strong constraint from the track, the track point data for the time k at the estimated track position s becomes train state data after a conversion to the train frame: $\mathcal{C}_{\text{track}}^{\text{train}}(\text{Tp}_s)$. The effective curvature and attitude for the train is interpolated from the map database and corrected by the train direction. The curvatures are translated directly by the sign of the train direction:

$$\begin{aligned} C_{\phi,k} &= dir_k \tfrac{d\phi}{ds}, \\ C_{\theta,k} &= dir_k \tfrac{d\theta}{ds}, \\ C_{\psi,k} &= dir_k \tfrac{d\psi}{ds}. \end{aligned} \tag{12}$$

The track attitude angles are converted to the train frame by:

$$\begin{aligned} \phi_k &= dir_k \cdot \phi_s, \\ \theta_k &= dir_k \cdot \theta_s, \\ \psi_k &= \begin{cases} \psi_s & \text{if } dir_k \text{ is positive} \\ \psi_s + \pi & \text{if } dir_k \text{ is negative} \end{cases} \end{aligned} \tag{13}$$

Finally, train states contain the train velocity and acceleration, the topological train position estimate, and track point data from that position converted into the train frame:

$$\begin{aligned} \text{T}_k = \{ & \text{P}_k^{\text{topo}}, \dot{s}_k, \ddot{s}_k, \mathcal{C}_{\text{track}}^{\text{train}}(\text{Tp}_s)\} = \\ \{ & \underbrace{R, s, dir}_{\text{topo position}}, \underbrace{\dot{s}, \ddot{s}}_{\text{train motion}}, \underbrace{lat, lon, alt}_{\text{geo position}}, \underbrace{\phi, \theta, \psi}_{\text{attitude}}, \underbrace{C_\phi, C_\theta, C_\psi}_{\text{curvatures}}\}_k. \end{aligned} \tag{14}$$

B. Intrinsic model for train kinematics

The geometry of a track influences the train states over the rail constrains. The geometry and train speed causes turn rates and centripetal accelerations as follows:

$$\dot{\psi} = \frac{d\psi}{dt} = \frac{d\psi}{ds} \cdot \frac{ds}{dt} = \frac{d\psi}{ds} \dot{s}, \tag{15}$$

$$a_{\text{cross}} = \dot{\psi}\dot{s} = \underbrace{\frac{d\phi}{ds}}_{\text{map}} \cdot \dot{s}^2. \tag{16}$$

The remaining turn rates $\dot{\phi}, \dot{\theta}$ and the centripetal acceleration a_{vertical} are calculated analogically. In a previous work [10] we defined a kinematic model based on the train speed and acceleration $\dot{s}, \ddot{s}$, the curvatures C_ϕ, C_θ, C_ψ and the track attitude ϕ, θ, ψ. For a self-contained publication we repeat the definitions. The train frame accelerations are dependent on the gravity vector, the train traction acceleration $\ddot{s}$ and the centripetal force corrected by rotations from the attitude:

$$\begin{aligned} a^x &= g \sin\theta + \overbrace{\ddot{s}}^{\text{traction}}, & (17) \\ a^y &= -g \sin\phi \cos\theta + C_\psi \dot{s}^2 \cos\phi - C_\theta \dot{s}^2 \sin\phi, & (18) \\ a^z &= \underbrace{-g \cos\phi \cos\theta}_{\text{gravity}} - \underbrace{C_\psi \dot{s}^2 \sin\phi}_{\text{cross centripetal acceleration}} - \underbrace{C_\theta \dot{s}^2 \cos\phi}_{\text{vertical centripetal acceleration}}. & (19) \end{aligned}$$

The train frame turn rates depend on the geometry, speed and attitude:

$$\begin{aligned} \omega^x &= C_\phi \dot{s} - C_\psi \dot{s} \sin\theta, & (20) \\ \omega^y &= C_\theta \dot{s} \cdot \cos\phi + C_\psi \dot{s} \sin\phi \cos\theta, & (21) \\ \omega^z &= \underbrace{C_\psi \dot{s}}_{\text{major turn rates}} \underbrace{\cdot \cos\theta \cos\phi - C_\theta \dot{s} \sin\phi}_{\text{attitude corrections}}. & (22) \end{aligned}$$

C. Rail Vehicle Filter

The train state is estimated with an Extended Kalman Filter (EKF). As topological pose contains the discrete track ID, only a delta distance Δs is estimated by the EKF. The EKF is wrapped by a train state estimation class, which contains the RailSLAM-phase state machine and the full topological pose. The geographic position is estimated in the EKF by metric differences in the local navigation frame by north ΔN, east ΔE and down ΔD, due to the non-linearity of the scaling of latitude.

1) System model: The EKF state μ_k contains 12 state variables and a nonlinear system model $f(\mu_{k-1}, \nu_{k-1})$ with process noise ν_k [14]:

$$\mu_k = f(\mu_{k-1}, \nu_{k-1}) \tag{23}$$

The full state and system model with the process noise of acceleration $\nu_{\ddot{s}}$ and curvatures ν_{C_ϕ}, ν_{C_θ} and ν_{C_ψ} is:

$$\begin{pmatrix} \Delta s \\ \dot{s} \\ \ddot{s} \\ \Delta N \\ \Delta E \\ \Delta D \\ \phi \\ \theta \\ \psi \\ C_\phi \\ C_\theta \\ C_\psi \end{pmatrix}_k = \begin{pmatrix} \dot{s}_{k-1}\Delta t + \ddot{s}_{k-1}\frac{\Delta t^2}{2} \\ \dot{s}_{k-1} + \ddot{s}_{k-1}\Delta t \\ \ddot{s}_{k-1} + \nu_{\ddot{s},k-1} \\ (\dot{s}_{k-1}\Delta t + \ddot{s}_{k-1}\frac{\Delta t^2}{2}) \cos(\theta_{k-1}) \cos(\psi_{k-1}) \\ (\dot{s}_{k-1}\Delta t + \ddot{s}_{k-1}\frac{\Delta t^2}{2}) \cos(\theta_{k-1}) \sin(\psi_{k-1}) \\ -(\dot{s}_{k-1}\Delta t + \ddot{s}_{k-1}\frac{\Delta t^2}{2}) \sin(\theta_{k-1}) \\ \phi_{k-1} + C_{\phi,k-1}\dot{s}_{k-1} \\ \theta_{k-1} + C_{\theta,k-1}\dot{s}_{k-1} \\ \psi_{k-1} + C_{\psi,k-1}\dot{s}_{k-1} \\ C_{\phi,k-1} + \nu_{C_\phi,k-1} \\ C_{\theta,k-1} + \nu_{C_\theta,k-1} \\ C_{\psi,k-1} + \nu_{C_\psi,k-1} \end{pmatrix}. \tag{24}$$

2) Extended Kalman Filter: The EKF predicts the state μ and the covariance Σ by a Kalman filter and linearizes the system model with a first order Taylor series by Jacobian matrices $F_{\mu,k} = \frac{df(\mu)}{d\mu}|_{\mu=\mu_k}$ and for the process noise $F_{\nu,k} = \frac{df(\nu)}{d\nu}|_{\nu=\nu_k}$. With process noise variance as a diagonal matrix $Q_k = diag(\sigma_{\ddot{s}}^2, \sigma_{C_\phi}^2, \sigma_{C_\theta}^2, \sigma_{C_\psi}^2)$, the prediction step is [14]:

$$\begin{aligned} \mu_{k|k-1} &= f(\mu_{k-1}), \\ \Sigma_{k|k-1} &= F_{\mu,k-1}\Sigma_{k-1|k-1}F_{\mu,k-1}^{\text{T}} + F_{\nu,k-1}Q_{k-1}F_{\nu,k-1}^{\text{T}}. \end{aligned} \tag{25}$$

The general measurement update with the gain K is computed with the linearized measurement matrix H and the sensor noise R [14]:

$$
\begin{aligned}
K_k &= \Sigma_{k|k-1} \cdot H_k^{\mathrm{T}} \cdot (H_k \Sigma_{k|k-1} H_k^{\mathrm{T}} + R_k)^{-1}, \\
\mu_{k|k} &= \mu_{k|k-1} + K_k \cdot (z_k - h(\mu_{k|k-1})), \\
\Sigma_{k|k} &= (I - K_k H_k) \Sigma_{k|k-1} (I - K_k H_k)^{\mathrm{T}} + K_k R_k K_k^{\mathrm{T}},
\end{aligned} \tag{26}
$$

where I is an identity matrix, h the measurement model and the covariance update is in the Joseph form [14].

3) GNSS measurement model: A GNSS measurement contains the position in LLA coordinates, absolute value of vehicle speed and heading in motion direction of the receiver antenna. The rail vehicle filter computes only metric position differences in north, east and down direction and speed and heading in the train frame. The metric differences of ΔN, ΔE and ΔD are calculated by appropriate transformations [2] from the difference of the position measurement $\mathrm{Z}_k^{\text{(lat, lon, alt)}}$ and the previous train state $\mathrm{T}_{k-1}^{\text{(lat, lon, alt)}}$ in LLA coordinates. The speed and heading measurements are transformed in the train frame depending on forward or backward motion of the train. In the case of a backward motion, the speed measurement is multiplied by (-1) and the heading measurement is shifted by π. These transformations result in $\hat{Z}_k^{\text{GNSS}} = (\Delta N, \Delta E, \Delta D, \dot{s}, \psi)_k^{\top}$ as a modified measurement. The GNSS measurement model is defined as:

$$
\hat{Z}_k^{\text{GNSS}} = \mathrm{H}^{\text{GNSS}} \cdot \mu_k + \left(\nu^N, \nu^E, \nu^D, \nu^{\dot{s}}, \nu^{\psi}\right)_k^{\top} \tag{27}
$$

The linear matrix H^{GNSS} is a 5x12 matrix selecting the speed $\dot{s}$, ΔN, ΔE, ΔD and the heading ψ. The measurement noise matrix is $R^{\text{GNSS}} = diag(\sigma_N^2, \sigma_E^2, \sigma_D^2, \sigma_{\dot{s}}^2, \sigma_{\psi}^2)$. After the update step, the relative ΔN, ΔE and ΔD coordinates of μ_k are added to the previous position $\mathrm{T}_{k-1}^{\text{(lat, lon, alt)}}$ by suitable transformations [2].

4) IMU measurement model: A nonlinear measurement model maps the acceleration and turn rate measurements to the train state.

$$
\mathrm{Z}_k^{\text{acc}} = \mathrm{h}^{\text{IMU}}(\mu_k) + (\nu^{ax}, \nu^{ay}, \nu^{az}, \nu^{\omega x}, \nu^{\omega y}, \nu^{\omega z})_k^{\top}, \tag{28}
$$

The nonlinear model $\mathrm{h}^{\text{IMU}}(\mu_k)$ is a vector of the six kinematic equations (17)-(22), and is linearized with a first order Taylor series by a Jacobian matrix $H_k^{\text{IMU}} = \frac{dh^{\text{IMU}}(z)}{dz}\Big|_{z=z_k^{\text{IMU}}}$. The IMU measurement noise matrix is $R^{\text{IMU}} = diag(\sigma_{\text{accX}}^2, \sigma_{\text{accY}}^2, \sigma_{\text{accZ}}^2, \sigma_{\text{gyroX}}^2, \sigma_{\text{gyroY}}^2, \sigma_{\text{gyroZ}}^2)$.

D. Map update

In the white space mapping phase, a track point is generated after reaching a minimum distance (e.g. 5m). This distance is calculated by the integration of the EKF output Δs. The track point Tp_s is computed from the actual train state T_k corrected by the train direction (12)-(13) and the variances from the covariance matrix diagonal Σ_k.

VII. PROOF OF CONCEPT

GNSS and IMU data was recorded with a rail vehicle at an industrial railway track near Braunschweig, Germany. The recorded sensor data of a Septentrio PolaRx3 GNSS receiver is processed at a 1Hz data rate and the XsensMTx MEMS IMU with 10Hz.

Figure 7 shows the probabilistic map, consisting of a single track of 300m length. All geometry data and their σ-deviations from the diagonal of the covariance matrix Σ are a function of the track position s. The third graph contains the

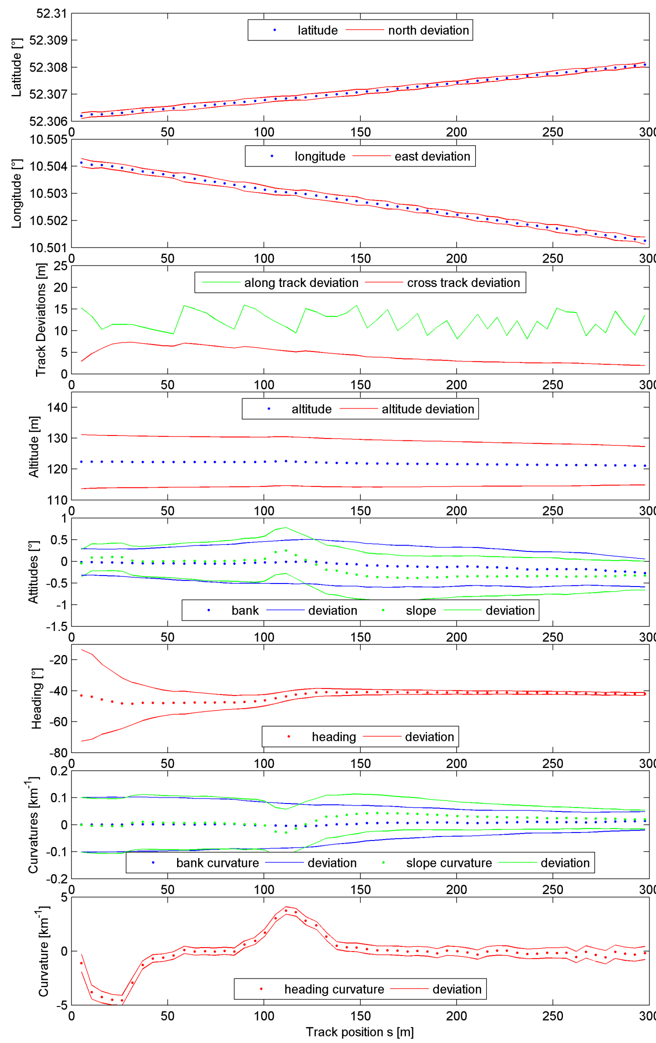

Fig. 7. Probabilistic geometric feature-rich map: Track geometry and deviations over track position s of a single track.

geometric track deviations in cross and along axis. The cross axis shows a lower deviation because the rail vehicle filter model of (24) restricts lateral movement. On the other hand, the filter model predicts motion in the along axis direction, which results in higher deviations. The along axis error varies because after IMU updates, the along deviation grows, and after GNSS updates the deviation is reduced. The mapping process is triggered every 5 meter and not by time. This method shows good reproducibility for the same tracks with different runs and data sets.

VIII. SUMMARY AND CONCLUSIONS

We have defined a concept of a SLAM approach for railways, based on a general Bayesian theory for a railway specific SLAM filter. The rail vehicle filter combines GNSS with IMU sensor data and estimates the train position, speed and acceleration as well as the track geometry by the attitude, geographic positions and three dimensional curvatures including deviations. Here, we focused on the mapping of a probabilistic feature-rich map, in which the track geometry data is parametrized on the topological track position.

The rail vehicle filter is a fundamental part for geometric map generation which enables localization based on geometric track features. In [10] we demonstrated the importance of geometry for a track selective localization assuming the existence of a sufficiently accurate map. In this paper we developed the *probabilistic geometric feature-rich map* which is a key component for localization and SLAM.

So far, the unknown data association of tracks R and track positions s is not addressed here. Promising filter techniques are Rao-Blackwellized particle filters as used in FastSLAM [15]. There, particles resolve the data association and Kalman filters compute the landmarks. In the future, we will focus on the combination of our train localization method based on geometric track features [10] and the presented method for the generation of a probabilistic map with topology and geometry information. Particles will contain the RailSLAM-phase state machine and a rail vehicle filter for the vehicle state and map estimation.

IX. ACKNOWLEDGMENTS

We thank Adrián Cardalda García for his work for the initial software framework, Omar García Crespillo for the sensor data parser (both former DLR), Andreas Lehner and the DLR RCAS2 project team for the measurements and Michael Angermann (DLR) for the fruitful discussions.

X. APPENDIX: DERIVATION OF FACTORIZATION

The operators of this derivation are Bayesian rule, product rule, conditional independence and the Markov assumption. The equation is proportional ($\propto$) when a normalization ignores a constant factor which can be simply calculated, as all probabilities have to sum up to one. The localization posterior of (2), is derived in a recursive factorization:

$$
\begin{aligned}
&p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k}) = \\
&\overset{\text{Bayes}}{=} \frac{p(\mathbf{Z}_k|\{\mathbf{TEU}\}_{0:k},\mathbf{Z}_{1:k-1})\cdot p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1})}{p(\mathbf{Z}_k|\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1})} \\
&\overset{\text{norm.}}{\propto} p(\mathbf{Z}_k|\{\mathbf{TEU}\}_{0:k},\mathbf{Z}_{1:k-1})\cdot p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1}) \\
&\overset{\substack{\text{cond.indep.}\\ \text{Markov}}}{\propto} \quad p(\mathbf{Z}_k|\{\mathbf{TE}\}_k)\cdot p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k}\mathbf{Z}_{1:k-1})
\end{aligned} \tag{29}
$$

The first factor of (29) is splits in the two types of $\mathbf{Z}$:

$$
\begin{aligned}
p(\mathbf{Z}_k|\{\mathbf{TE}\}_k) &= p(\mathbf{Z}_k^{\text{IN}},\mathbf{Z}_k^{\text{EX}}|\mathbf{T}_k,\mathbf{E}_k^{\text{IN}},\mathbf{E}_k^{\text{EX}}) = \\
&\overset{\substack{\text{prod. rule}\\ \text{cond.indep.}}}{=} \quad p(\mathbf{Z}_k^{\text{IN}}|\mathbf{T}_k,\mathbf{E}_k^{\text{IN}})\cdot p(\mathbf{Z}_k^{\text{EX}}|\mathbf{T}_k,\mathbf{E}_k^{\text{EX}}).
\end{aligned} \tag{30}
$$

The second factor of (29) is factored in a recursive form:

$$
\begin{aligned}
&p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1}) = \\
&\overset{\text{prod. rule}}{=} \quad p(\{\mathbf{TE}\}_k|\{\mathbf{TE}\}_{0:k-1},\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1})\cdot \\
&\qquad \cdot p(\{\mathbf{TE}\}_{0:k-1}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k-1}) \\
&\overset{\text{cond.indep., Markov}}{=} p(\{\mathbf{TE}\}_k|\{\mathbf{TE}\}_{k-1},\mathbf{U}_k)\cdot \\
&\qquad \cdot \underbrace{p(\{\mathbf{TE}\}_{0:k-1}|\mathbf{U}_{0:k-1}\mathbf{Z}_{1:k-1})}_{\text{recursive part: posterior of k-1}}.
\end{aligned} \tag{31}
$$

The first factor of (31) is factorized again:

$$
\begin{aligned}
&p(\{\mathbf{TE}\}_k|\{\mathbf{TE}\}_{k-1},\mathbf{U}_k) = \\
&\overset{\text{prod. rule}}{=} p(\mathbf{T}_k|\{\mathbf{TE}\}_{k-1},\mathbf{E}_k,\mathbf{U}_k)\cdot p(\mathbf{E}_k|\{\mathbf{TE}\}_{k-1},\mathbf{U}_k) \\
&\overset{\text{cond.indep.}}{=} p(\mathbf{T}_k|\mathbf{T}_{k-1},\mathbf{U}_k)\cdot p(\mathbf{E}_k|\mathbf{E}_{k-1})
\end{aligned} \tag{32}
$$

Inserting (32) in (31) together with (30) in (29) results in the factorized localization posterior:

$$
\begin{aligned}
&p(\{\mathbf{TE}\}_{0:k}|\mathbf{U}_{0:k},\mathbf{Z}_{1:k}) \propto \\
&\propto p(\mathbf{Z}_k^{\text{IN}}|\mathbf{T}_k,\mathbf{E}_k^{\text{IN}})\cdot p(\mathbf{Z}_k^{\text{EX}}|\mathbf{T}_k,\mathbf{E}_k^{\text{EX}})\cdot p(\mathbf{T}_k|\mathbf{T}_{k-1},\mathbf{U}_k)\cdot \\
&\quad \cdot p(\mathbf{E}_k|\mathbf{E}_{k-1})\cdot p(\{\mathbf{TE}\}_{0:k-1}|\mathbf{U}_{0:k-1}\mathbf{Z}_{1:k-1})
\end{aligned} \tag{33}
$$

REFERENCES

[1] T. Strang, M. Meyer zu Hoerste, and X. Gu, “A Railway Collision Avoidance System Exploiting Ad-hoc Inter-vehicle Communications and Galileo,” in *13th World Congress on Intelligent Transportation Systems, London, UK*, 2006.

[2] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2006.

[3] International Union of Railways, “RMTS - European Rail Management Traffic System,” http://www.uic.org/spip.php?rubrique847.

[4] M. Angermann and P. Robertson, “FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors - Hitchhiking on Human Perception and Cognition,” in *Proceedings of the IEEE*, may 2012.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.

[6] C. Hasberg, S. Hensel, and C. Stiller, “Simultaneous Localization and Mapping for Path-Constrained Motion,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –12, 2011.

[7] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995, iSBN: 978-0131038059.

[8] S. Hensel, C. Hasberg, and C. Stiller, “Probabilistic Rail Vehicle Localization With Eddy Current Sensors in Topological Maps,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1525 –1536, dec. 2011.

[9] A. Cardalda Garcia, “RailSLAM: Simultaneous Localization and Mapping for Railways,” Master’s thesis, University of Oviedo, Spain, 2012.

[10] O. Heirich, P. Robertson, A. Cardalda Garcia, and T. Strang, “Bayesian Train Localization Method Extended by 3D Geometric Railway Track Observations From Inertial Sensors,” in *ISIF Information Fusion Conference*, july 2012.

[11] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology, 2nd Edition*. The Institution of Electrical Engineers (IEE), 2004.

[12] C. Hasberg and S. Hensel, “Bayesian mapping with probabilistic cubic splines,” in *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, 29 2010-sept. 1 2010.

[13] *Railway safety performance in the European Union 2012*. European Railway Agency, 2011. [Online]. Available: http://www.era.europa.eu

[14] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, 2001.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.